智慧安全帽

組員 ： 何承睿、何秉翰、黃毓宸、溫冠閔
指導老師 ： 簡靖哲老師

組員介紹

何承睿

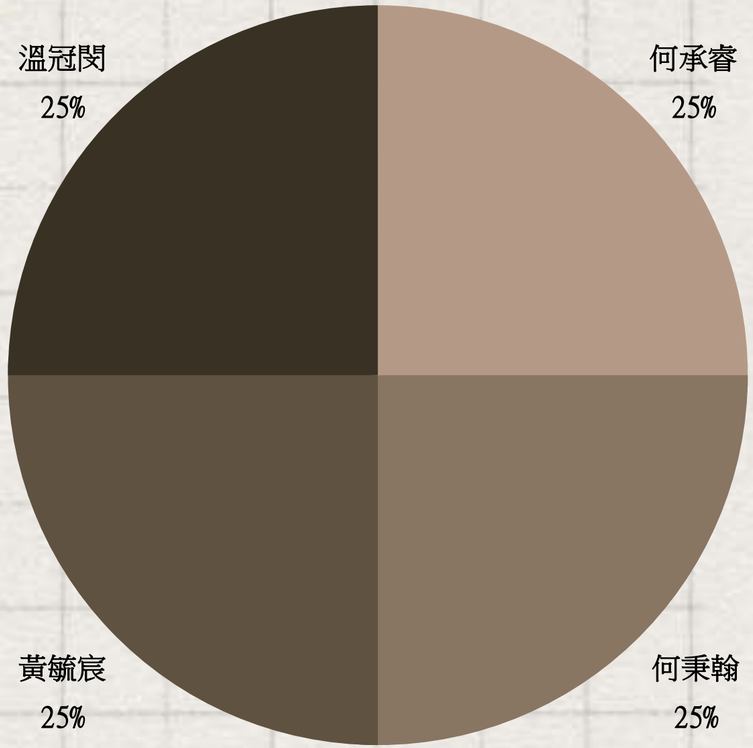
何秉翰

黃毓宸

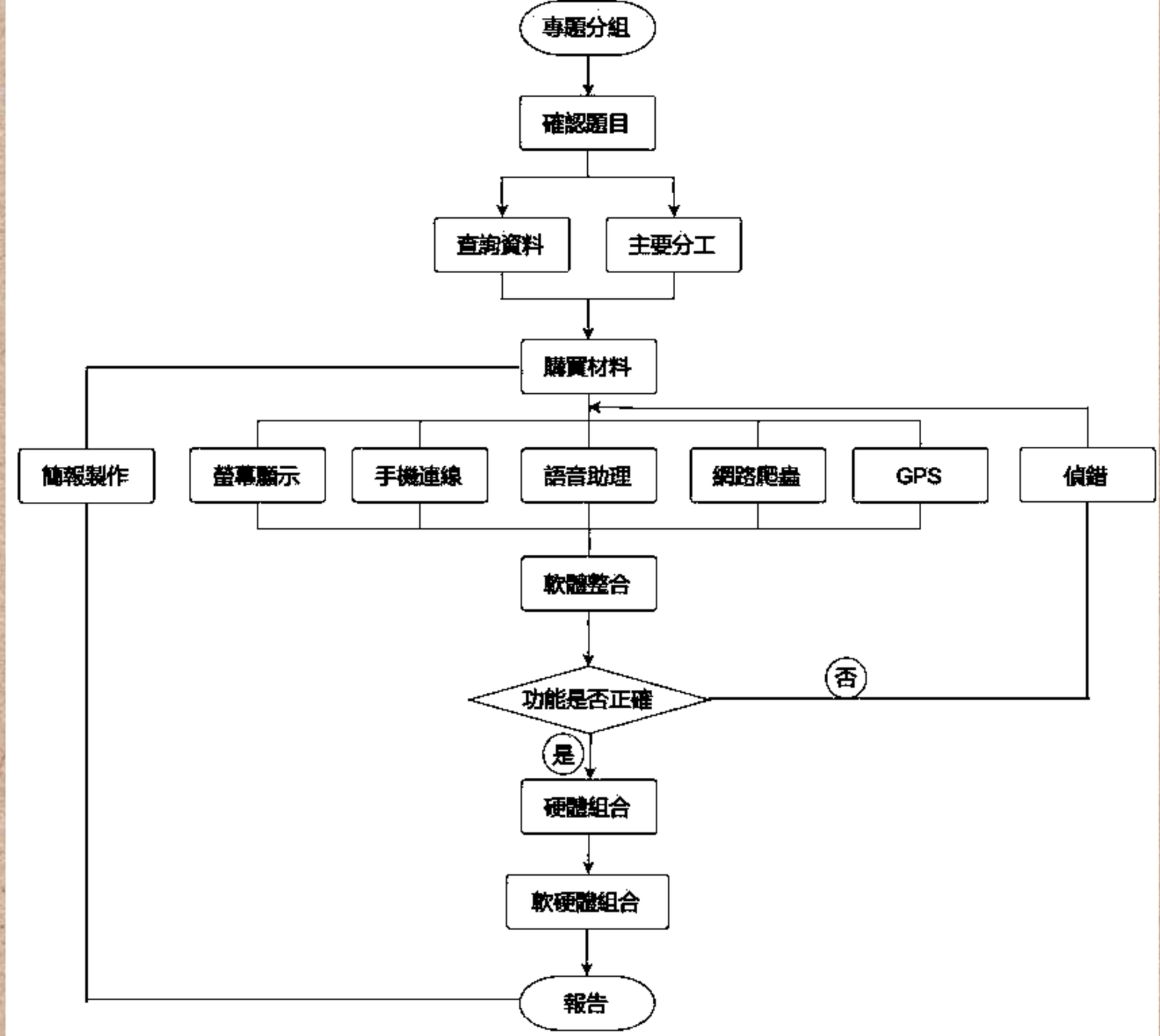
溫冠閔

分工

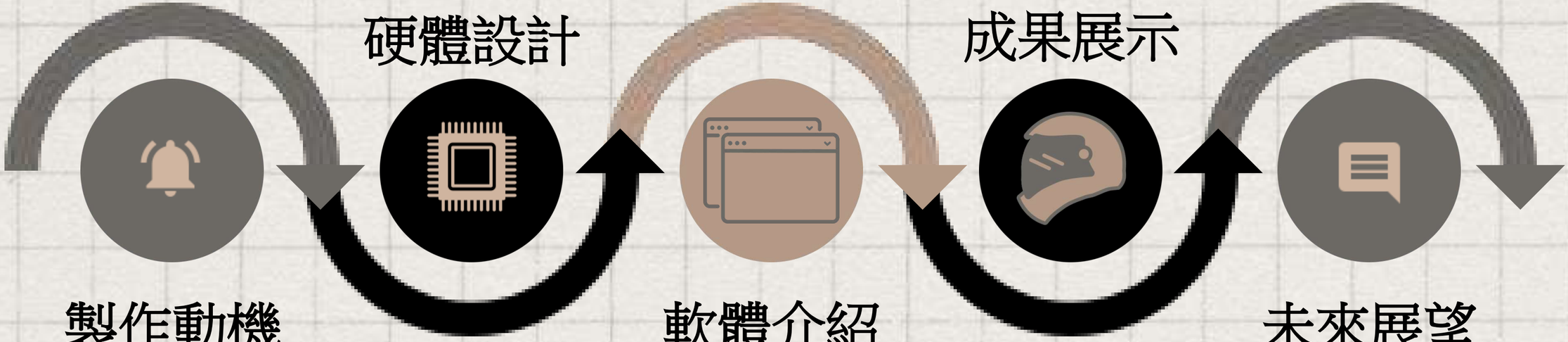
何承睿	GOOGLE助理
何秉翰	天氣爬蟲、TFT螢幕
黃毓宸	GPS模組、簡報美化
溫冠閔	LINE NOTIFY、外殼製作



流程圖


專題分組
確認題目
查詢資料
主要分工
購買材料
簡報製作
螢幕顯示
手機連線
語音助理
網路爬蟲
GPS
偵錯
軟體整合
功能是否正確
否
是
硬體組合
軟硬體組合
報告

專題總覽



製作動機

新北市1名男子，日前駕駛汽車行經路口時，與1名機車騎士發生擦撞，未料事後騎士一口就要求償20萬元，讓他覺得很不合理，並氣得將行車紀錄器PO網，指控騎士邊騎車邊看手機，本身要負7成責任。

預期功能

天氣

利用網路爬蟲截取天氣資訊，並且用LINE NOTIFY傳送到手機上。再透過TFT螢幕顯示到眼前。

GPS

利用GPS模組讀取地理資訊，並將經緯度與Google 助理共享、轉換成地址傳送至LINE NOTIFY

GOOGLE助理

利用Google Assistant和谷歌小姐對話並利用GPS模組提供的經緯度使用導航功能

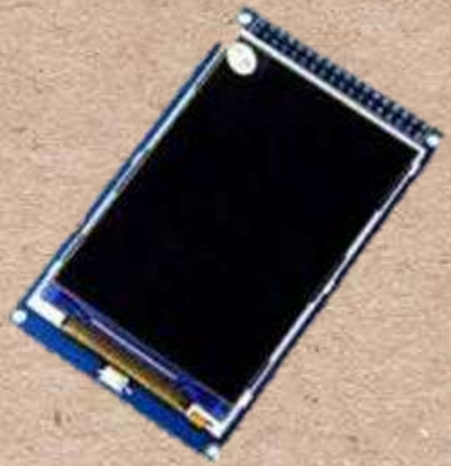
硬體介紹

樹莓派

是一款基於Linux系統的單板機電腦，體積只有一張信用卡的大小

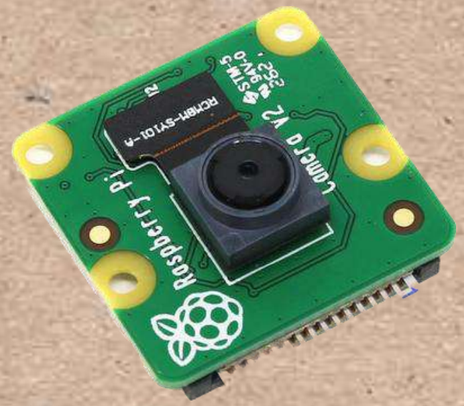
GPS模組

讀取頻率: 10Hz
特性: 開機速度快
消耗功率小



TFT螢幕

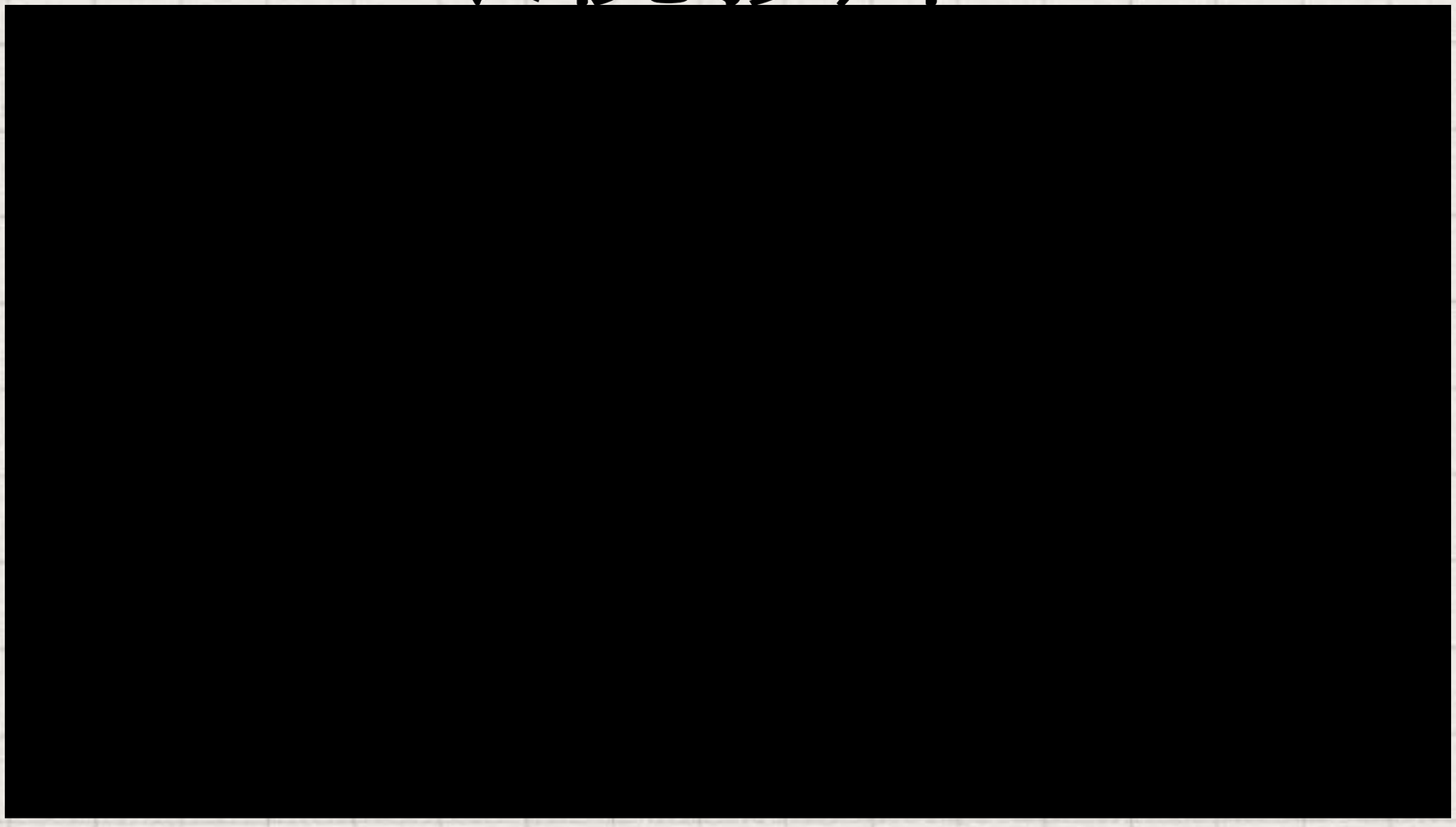
驅動IC: ST7735
彩色TFT LCD 65K色
解析度: 128x160 點



相機模組

鏡頭 : 1/4 5M
光圈 : 2.9
焦距 : 3.29
視場角 : 72.4度

環視影片



LINE NOTIFY

LINE NOFIFY 是一種免費並能夠自動發送Line訊息的工具

LINE NOTIFY

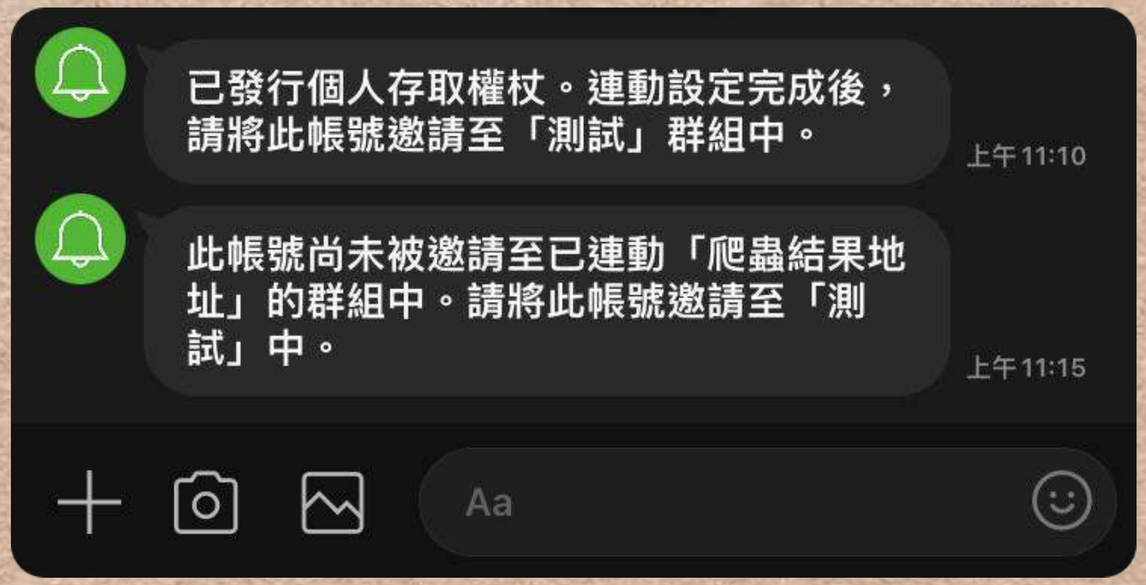
先到LINE NOFIFY網站取得權杖接著連動服務

LINE NOTIFY

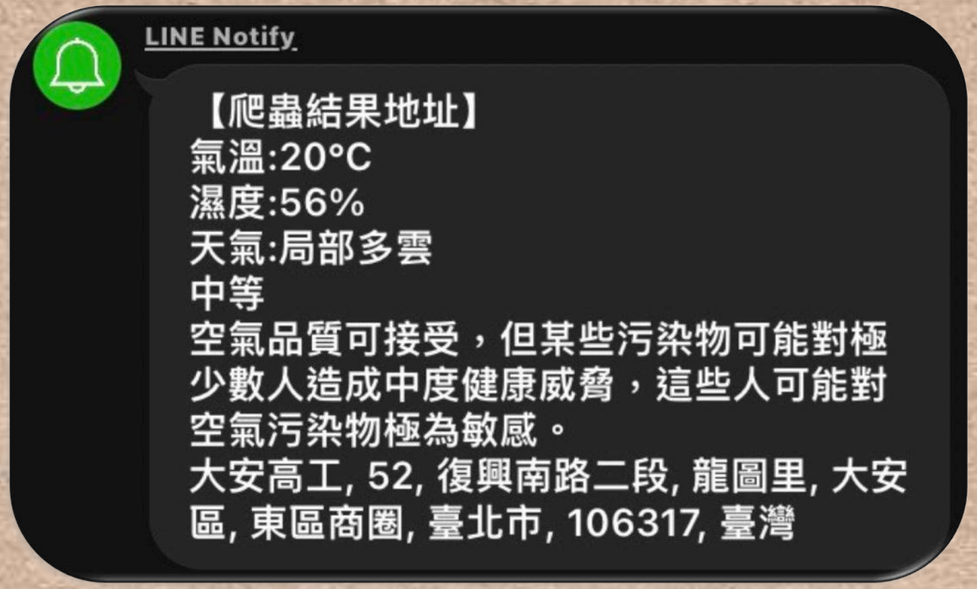
```python
import requests

def lineNotifyMessage(token, msg):
        headers = {
                "Authorization" : "Bearer " + token,
                "Content-Type" : "application/x-www-form-urlencoded"
        }

        payload = {'message': msg}
        r = requests.post("https://notify-api.line.me/api/notify", headers = headers, params = payload)
        return r.status_code

if __name__ == "__main__":
    token = '權杖內容'
    message = '結果'
    lineNotifyMessage(token, message)
#結果為爬蟲結果和經緯度地址轉換的結果
```

將取得的權杖輸入程式碼中

LINE NOTIFY



將LINE NOTIFY邀請至群組



就可以收到樹梅派傳送的訊息！

Python 爬蟲

爬蟲流程

搜尋網站
尋找適合的網站擷取資料

取得使用者資訊
將使用者代理（user-agent)記入程式

尋找資料
確認資料在網頁中的編號

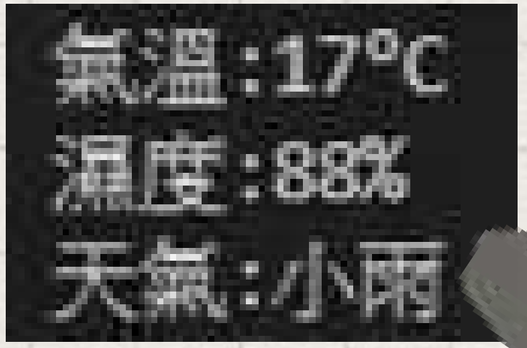
程式抓取資料
將網頁的原始碼擷取下來

篩選資料
在原始碼中取出需要的資料

輸出結果

```
sec-fetch-site: cross-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr
ome/104.0.0.0 Safari/537.36
```


```
onditions--primary--2DOqs">
"TemperatureValue" class="CurrentConditions--tempValue--MHmYY">17°
wxPhrase" class="CurrentConditions--phraseValue--mZC_p">小雨</div>
```




```python
#用id 搜尋濕度

humidity=root.find('span',{"data-testid":"PercentageValue"})

#溫度
temp=root.find("span",class_="CurrentConditions--tempValue--MHmYY")
temp2=("氣溫:"+temp.string+"C")
print(temp2)



print("濕度:"+humidity.string)

#雨況
rains=root.find('div', {"data-testid":"wxPhrase"})
print("天氣:"+rains.string)
```

GPS模組

製作流程

套件安裝 → 讀取資料測試 → 尋找接腳位址

套件自啟動 → PYTHON程式 → 連動其他程式

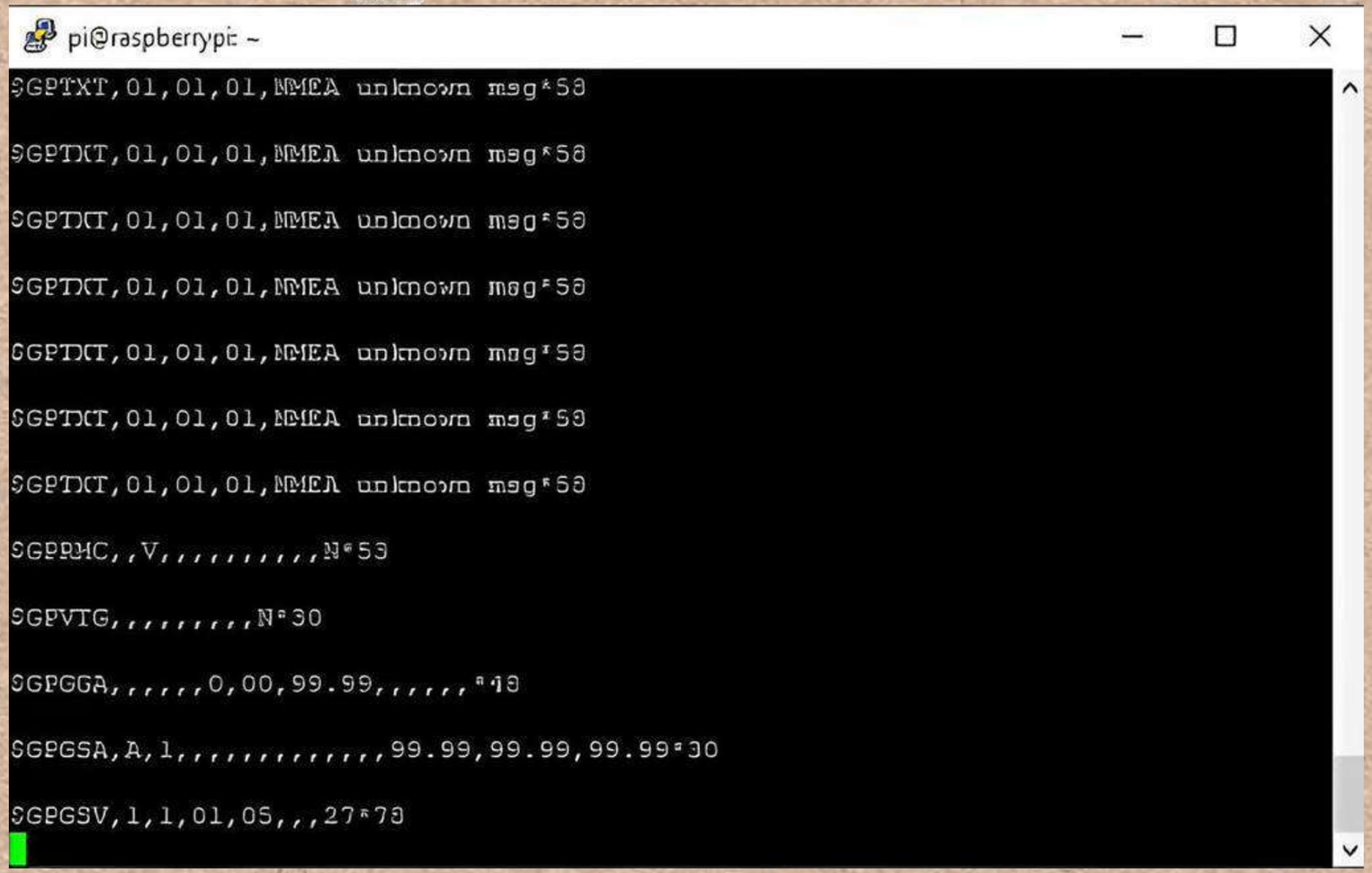
套件安裝

```
sudo apt-get install gpsd gpsd-clients
sudo pip install pynmea2
```

讀取資料測試

```
cat /dev/serial0
```

執行結果


pi@raspberrypi: ~
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPTXT,01,01,01,NMEA unknown msg*58
$GPRMC,,V,,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,,0,00,99.99,,,,,,*48
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,01,05,,,27*78

尋找接腳位址

尋找接腳位址

```
serial0 -> ttyAMA0
serial1 -> ttyS0
```

GPSD自啟動

```
sudo nano /etc/default/gpsd
```

GPSD自啟動

```
  GNU nano 5.4                                          /etc/default/gpsd
START_DAEMON="true"
# Devices gpsd should collect to at boot time.
# They need to be read/writeable, either by user gpsd or the group dialout.
DEVICES="/dev/ttyAMA0"

# Other options you want to pass to gpsd
GPSD_OPTIONS="/dev/ttyAMA0"
GPSD_SOCKET="/var/run/gpsd.sock"
# Automatically hot add/remove USB GPS devices via gpsdctl
USBAUTO="true"
```

GPS測試

測試結果

PYTHON程式

```
from gps import *
import time
running = True

def getPositionData(gps) :
    nx = gpsd.next()
    if nx['class'] =="TPV" :
        latitude = getattr(nx, 'lat', "Unknown")
        longitude = getattr(nx, 'lon', "Unknown")
        print ("Your Position: lon = " + str(longitude) + ", lat = " + str(latitude))

gpsd = gps(mode=WATCH_ENABLE|WATCH_NEWSTYLE)

try:
    print ("Application Started!")
    while running :
        getPositionData(gpsd)
        time.sleep(1.0)

except (KeyboardInterrupt):
    running = False
    print('\n' "Application Closed!")
```

PYTHON程式

```
def getPositionData(gps) :
    nx = gpsd.next()
    if nx['class'] =="TPV" :
        latitude = getattr(nx, 'lat', "Unknown")
        longitude = getattr(nx, 'lon', "Unknown")
        print ("Your Position: lon = " + str(longitude) + ", lat = " + str(latitude))
```

副程式

```
nx = gpsd.next()
if nx['class'] == "TPV" :
```

執行結果

```
Python 3.9.2 (/usr/bin/python3)
>>> %Run long.py

  Application Started!
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
  Your Position: lon = 121.5427221, lat = 25.0329571
```

連動其他程式

天氣爬蟲

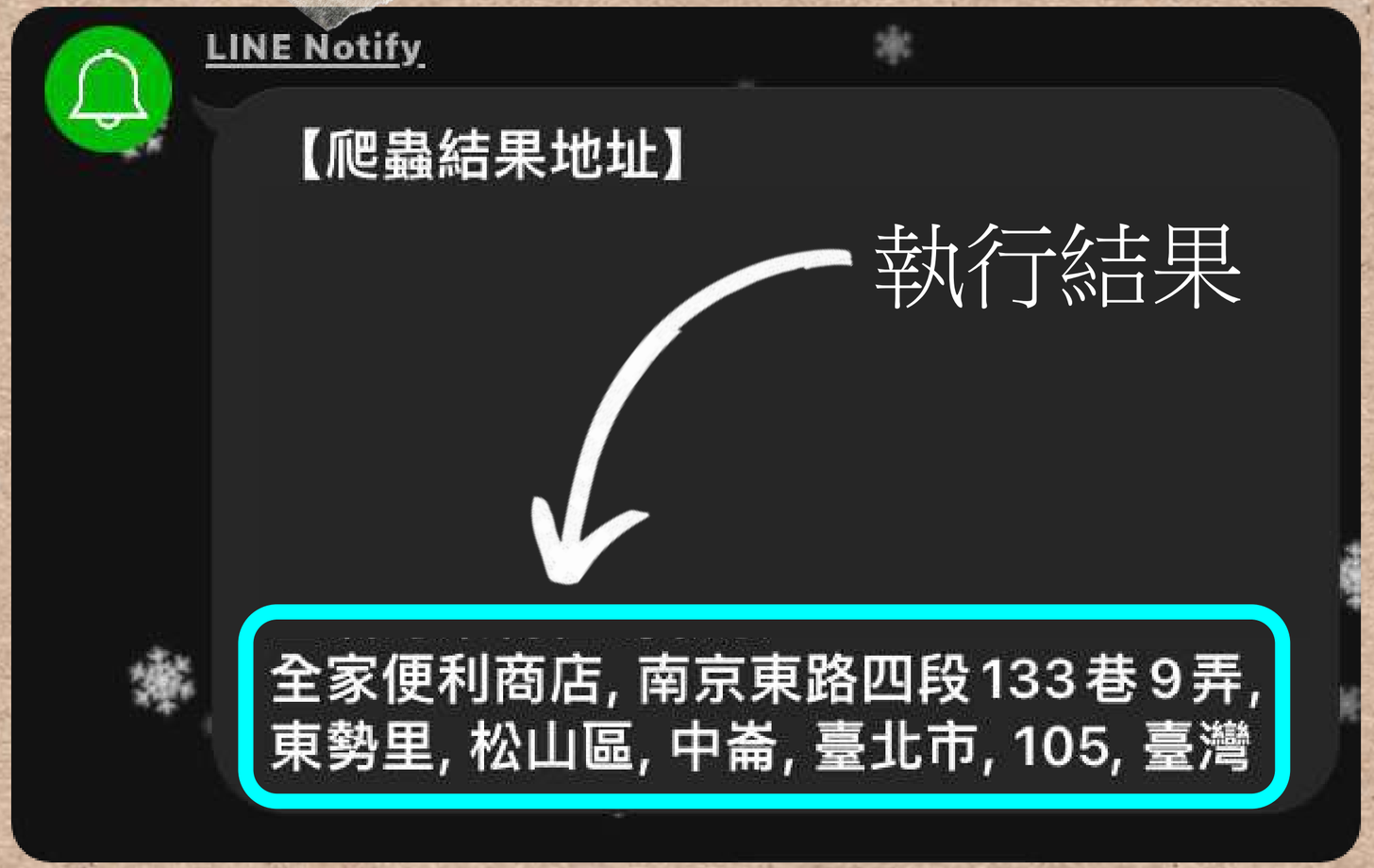
GOOGLE MAP

經緯度地址轉換

經緯轉換

```
from geopy.geocoders import Nominatim

geolocation = Nominatim(user_agent="geotest")

location = geolocation.reverse("緯度, 經度")
print(location.address)
}
```

經緯轉換


LINE Notify
【爬蟲結果地址】
執行結果
全家便利商店, 南京東路四段133巷9弄, 東勢里, 松山區, 中崙, 臺北市, 105, 臺灣

Google Assistant

GOOGLE助理

音頻設置

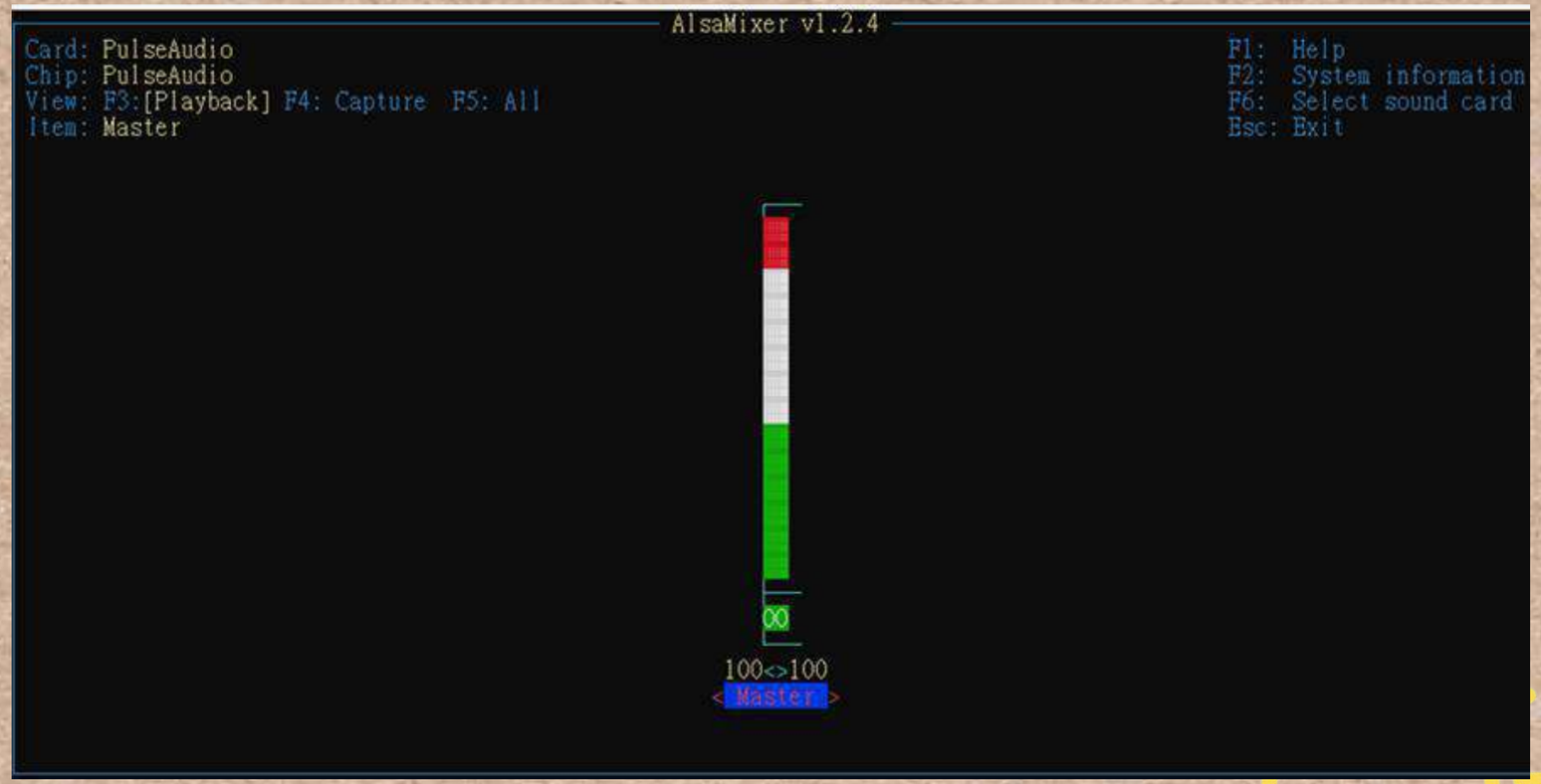
```
pi@pi:~ $ pi@pi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

卡號

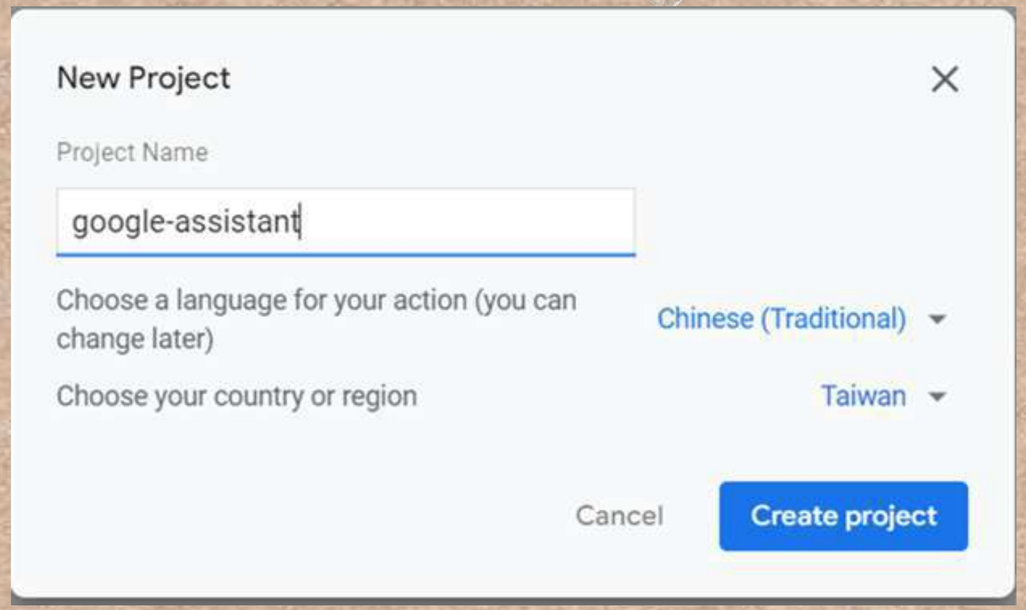
設備號

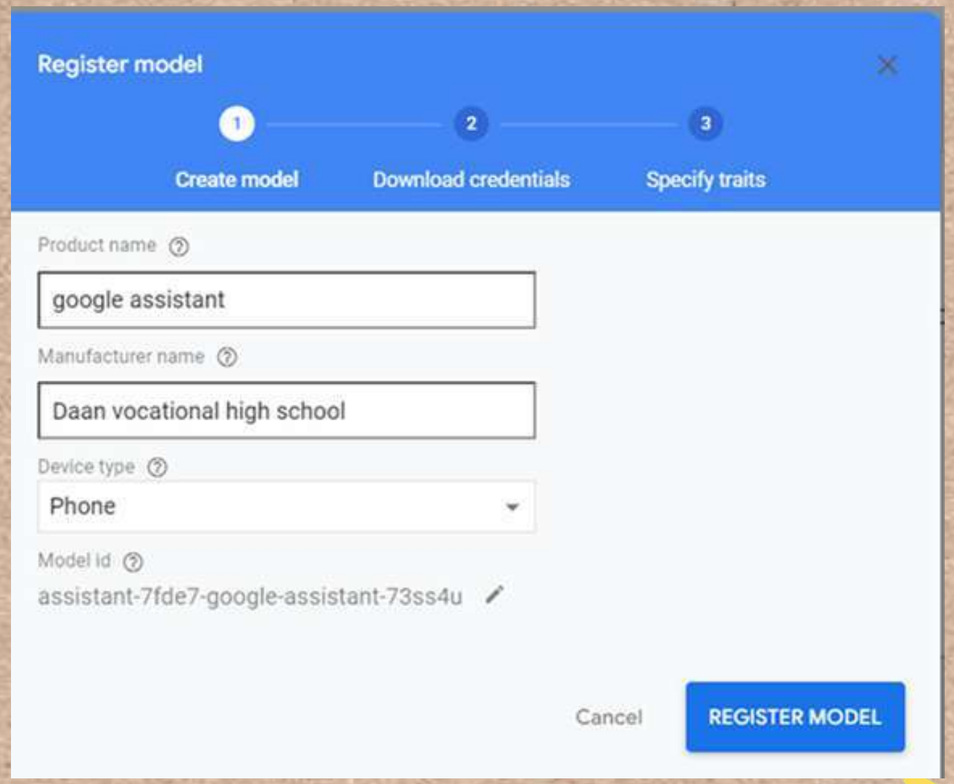
音頻設置

```
  GNU nano 5.4
pcm.!default {
  type asym
  capture.pcm "mic"
  playback.pcm "speaker"
}
pcm.mic {
  type plug
  slave {
    pcm "hw:1,0"
  }
}
pcm.speaker {
  type plug
  slave {
    pcm "hw:0,0"
  }
}
```

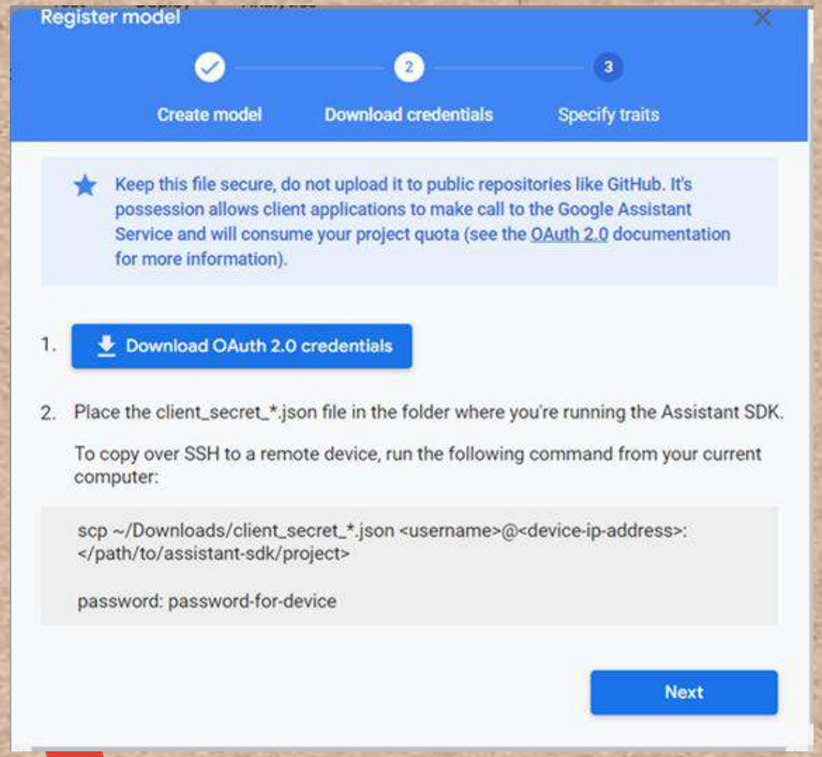


創建GOOGLE模型


New Project
Project Name
google-assistant
Choose a language for your action (you can change later)
Chinese (Traditional)
Choose your country or region
Taiwan
Cancel
Create project


Register model
1
Create model
2
Download credentials
3
Specify traits
Product name
google assistant
Manufacturer name
Daan vocational high school
Device type
Phone
Model id
assistant-7fde7-google-assistant-73ss4u
Cancel
REGISTER MODEL

下載憑證


Register model
Create model
2
Download credentials
3
Specify traits
Keep this file secure, do not upload it to public repositories like GitHub. It's possession allows client applications to make call to the Google Assistant Service and will consume your project quota (see the OAuth 2.0 documentation for more information).
1.
Download OAuth 2.0 credentials
2. Place the client_secret_*.json file in the folder where you're running the Assistant SDK.
To copy over SSH to a remote device, run the following command from your current computer:
scp ~/Downloads/client_secret_*.json <username>@<device-ip-address>:</path/to/assistant-sdk/project>
password: password-for-device
Next


C:\Windows\system32\cmd.exe

啟用 API & 啟用外部測試

後續設置

安裝虛擬環境

安裝SDK包

生成能夠運行的憑證

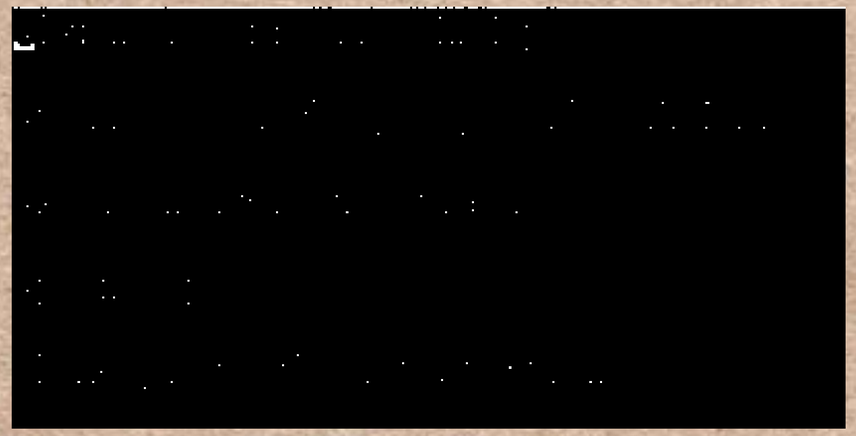
執行GOGLE助理結果

```
ON_MUTED_CHANGED:
  {"is_muted": false}

ON_CONVERSATION_TURN_STARTED
ON_END_OF_UTTERANCE
ON_END_OF_UTTERANCE
ON_RECOGNIZING_SPEECH_FINISHED:
  {"text": "這一週的天氣如何"}
ON_RESPONDING_STARTED:
  {"is_error_response": false}
ON_RESPONDING_FINISHED
ON_CONVERSATION_TURN_FINISHED:
  {"with_follow_on_turn": false}
```

谷歌助理自啟動

```
 GNU nano 5.4                    google_autostart.sh
#!/bin/sh

source env/bin/activate

/home/pi/env/bin/google-assistant-demo&
```

```
選取 pi@pi: ~/.config/lxsession/LXDE-pi
pi@pi:~ $ cd .config
session//LXDE-pi/
ls -lpi@pi:~/.config $ cd lxsession//LXDE-pi/
pi@pi:~/.config/lxsession/LXDE-pi $ ls -l
總用量 8
-rw-r--r-- 1 root root 133 10月 30 02:33 autostart
-rw-r--r-- 1 pi   pi   256 10月 29 23:09 desktop.conf
pi@pi:~/.config/lxsession/LXDE-pi $
```



遭遇問題

遭遇問題

我們在最後要整合系統的時候，

樹梅派突然無法正常開機，

導致程式無法執行，

上網查詢後研判應是SD卡的問題

所以我們將程式複製進新的記憶卡，

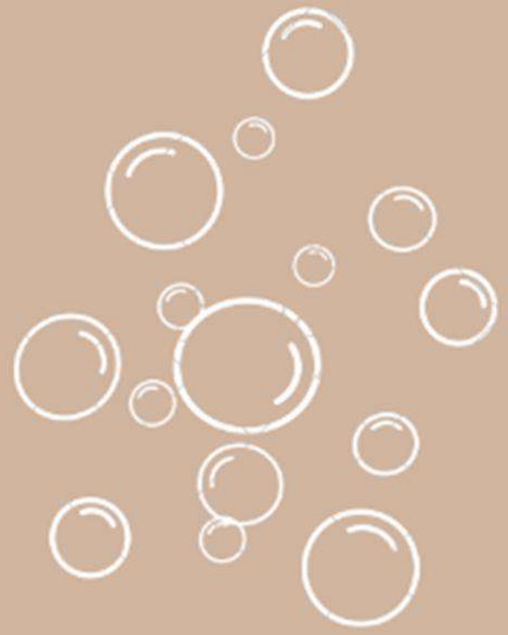
才處理完成。

成果展示

成果影片

案例一

未來展望



1. TFT顯示相機的畫面
2. 提高GPS模組精準度
3. TFT螢幕能夠顯示現在的位置
4. 用GPS模組計算速度並且顯示出來
5. 縮小設備體積、與安全帽整合

參考資料

https://www.youtube.com/watch?v=9Z9xKWfNo7k

https://www.youtube.com/watch?v=cta1yCb3vA8&t=391s

https://www.youtube.com/watch?v=wav1bHOM1Zg

https://www.factoryforward.com/raspberry-pi-google-assistant/

參考資料-GPS

https://raspberrypi.stackexchange.com/questions/29547

https://www.instructables.com/Interfacing-GPS-Module-With-Raspberry-Pi/

https://maker.pro/raspberry-pi/tutorial/how-to-use-a-gps-receiver-with-raspberry-pi-4

https://youtu.be/SWLNn1Ox9KY

THANK YOU

提問時間